An Open Experimental Platform for Ranging, Proximity and Contact Event Tracking using Ultra-Wide-Band and Bluetooth Low-Energy

Roudy Dagher, Francois-Xavier Molina, Alexandre Abadie, Nathalie Mitton, Emmanuel Baccelli

An Open Experimental Platform for Ranging, Proximity and Contact Event Tracking using Ultra-Wide-Band and Bluetooth Low-Energy

Roudy Dagher, Francois-Xavier Molina, Alexandre Abadie, Nathalie Mitton and Emmanuel Baccelli
Inria, France. Email: first.lastname@inria.fr

***Abstract*—The need for cheaper and more precise localisation techniques has recently amplified. The initial approach has been to roll out high-level software running on smartphones and leveraging Bluetooth proximity sensing. However this approach lacks both precision in terms of ranging, and flexibility in terms of experimental framework to fully explore alternative schemes for contact event tracing. In this context, we thus provide open-access nodes in an open-access experimental platform for ranging and proximity tracking, letting researchers tinker freely with the full software stack on a swarm of multi-radio, low-power devices based on cheap microcontrollers. We provide a tutorial on how to use the platform and open source code building blocks to to program the devices, bare-metal. We then report on initial measurements we have performed using the platform. Perspectives with our platform include applicability studies and comparative evaluation for a large variety of localisation schemes combining the use of Ultra-Wide Band and Bluetooth Low-Energy for better precision and smaller energy budgets – and the use of complementary mechanism guaranteeing privacy protection, able to run directly on-board cheap IoT microcontrollers.**

I. Introduction

As the COVID-19 pandemic spreads, a race to develop efficient and privacy-friendly contact tracing systems is taking place. Initial solutions have been rolled out based on the dominant smartphone ecosystems (Apple's iOS and Google's Android), and using Bluetooth distance ranging. The success of such solutions is hampered by different factors, mainly (i) purposely restricted access to low level software on most smartphones, making it difficult to study/exploit the full spectrum of algorithmic possibilities and (ii) inherently imprecise proximity detection based on Bluetooth alone, making it difficult to have less than meter precision, which is not adequate w.r.t. typical social distancing recommendations (e.g. minimum 2 meters distance).

In this paper, we address this problem by providing an open-access experimental platform for ranging and proximity tracking. The platform consists of both (a) hardware, in the shape of an instrumented test network and (b) software, in the shape of open source building blocks to build embedded firmwares from scratch. The platform provides researchers with convenient means to remotely access and fully program (bare-metal) a swarm of low-cost, low-power IoT devices offering both Ultra-Wide-Band (UWB) and/or Bluetooth (BLE) wireless communication capabilities.

Indeed, UWB has emerged as one of the main technologies used for more precise localization. Large bandwidth leads to high time resolution (short pulses) which allows for precisely time-stamping received signals. High time resolution and short wavelength strengthen it against multi-path fading and interference. The large bandwidth also allows for a high-bit-rate. All these properties allow to pinpoint a device location in real-time with an accuracy under 20cm.

These hardware characteristics are compatible both with the upcoming generation of smartphones – which have dual UWB and BLE – and with the typical requirements of cheap physical tokens – an attractive alternative to smartphones for contact tracing [1].

The main contributions of this paper are the following:

- we make available new open access nodes on the FIT IoT LAB platform;
- we provide a portable, and fully open source software stack that allows researchers to (re)program the entire embedded software, including low-level radio drivers and network stacks;
- we report on initial proximity measurements using the platform, using UWB and BLE and basic proximity estimation algorithms;
- we provide tutorial information on how to use the testbed, and how to reprogram proximity estimation with arbitrary algorithms, using the open source building blocks we provide.

We first briefly overview background and related work in Section 2, then we describe our platform in section 3. We then provide a quick tutorial guide for users of the platform. Finally, in section 4 we provide initial proximity and ranging measurements using the platform, before we conclude on future work and potential perspectives using our experimental platform.

II. Background & Related Work

In this section, we provide a bref overview of existing solutions and current practices on localisation based either on UWB or BLE and the experimental platforms enabling radiolocalization evaluation.

Localisation with UWB: Prior work on position estimation with UWB uses various approaches and parameters. Among these, we can mention received-signal-strength (RSS), time-of-arrival (TOA), time-difference-of-arrival (TDOA) & phase-difference-of-arrival (PDOA) [2]. TOA is most commonly used for Two-Way-Ranging (TWR) protocols such as single-sided-TWR (SS-TWR), symmetric-double-sided-TWR (SDS-TWR), two-message-TWR (2M-TWR), alternative-double-sided-TWR (AltDS-TWR) [3], asymmetric-double-sided-TWR (ADS-TWR) [4], symmetric-double-sided-TWR-multiple-ACK (SDS-TWR-MA) [5], etc. These different protocols offer trade-offs between estimation error, clock skew tolerance, message delay, message number, etc.

Localisation with BLE: Indoors localisation using Bluetooth has been an active subject of research because of its wide-spread availability, and its ultra low-power characteristics. However, accuracy is usually over 1m [6] makes it unreliable as a stand-alone technology for location services. The emergence of dual BLE + UWB devices are making possible combined approaches, making the best of leveraging simultaneously UWB's high accuracy and BLE power efficiency. Beyond BLE & UWB, other radio technologies such as WiFi, Zigbee(802.15.4) can also be used for contact-tracing & social-distancing purposes[7], [8].

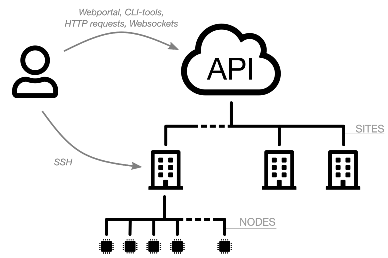
Experimental platforms for low-power localisation: Being able to experimentally evaluate combined multi-radio approaches on a common experimental platform is of high interest, as multi-radio devices become pervasive and research reproducible. Low-power devices can be conveniently programmed from scratch based on open source embedded software provided by Arduino sketches or by various embedded operating systems [9]. For instance, existing toolsets such as DecaDuino, Atlas & Wi-PoS[10] offer openSource UWB ranging hardware and software. In particular, DecaDuino devices are integrated into the LOCura testbed [11], but this facility is not in open access and focuses on UWB only. In contrast, the experimental platform we describe in this paper is in open access, and offers a multi-radio ranging facility.

III. Platform

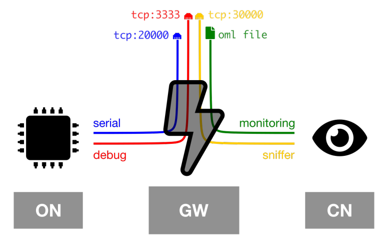
The platform consists in a testbed on which are deployed fully programmable IoT devices, including dual UWB/BLE devices, as well as open source software building blocks which can be used to program from scratch the IoT devices on the testbed.

A. Physical Testbed Deployment

The devices used to perform the experiments have been made available as an extension of the FIT IoT-LAB testbed [12] and physically deployed on the site of Saclay, France. This testbed provides an open access to a large scale and multi-sites deployment of heterogeneous IoT devices[1]. The infrastructure of the testbed[2] allows users to interact with the devices either via an API or directly from on-site front-end servers (Fig. 1). This flexible level of interaction offers a total control on the devices used during an experiment: users can fully (re)program the devices and, for each device, access the stdio serial port (via UART), a debugger (remotely with GDB), a radio sniffer (802.15.4 only) and a power consumption monitor. Within a single experiment, it's also possible to automatize complex networking scenarios with hundreds of devices across several sites.



(a) Global infrastructure



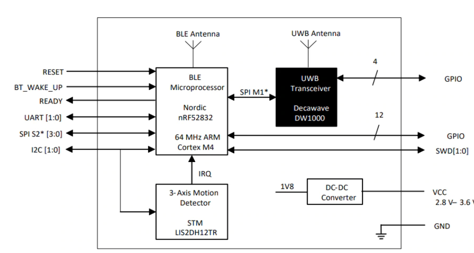
(b) Device management gateway

Fig. 1. (a) The global design of the FIT IoT-LAB testbed and (b) the management gateway (GW) used by the infrastructure to control each single open node (ON). The control node (CN) adds power consumption monitoring and radio sniffing capabilities.

FIT IoT-LAB was initially developed around IoT devices with 802.15.4 radio and recent deployments added access to devices with Sub-GHz, WiFi, LoRa, BLE and UWB radios. For example, the deployment on the Saclay site contains +40 devices with BLE radio, among Nordic nRF51DK/nRF52DK/nRF52840DK, Pycom FiPy and Decawave DWM1001.

B. UWB/BLE Open Node

(a) Board



(b) DWM1001 module

Fig. 2. The open DWM1001-DEV node embedding a BLE microprocessor, a UWB transceiver with their antennas and a 3-axis motion detector.

The DWM1001-DEV reference boards are based on the 802.15.4-2011 compliant DWM1001 module that exposes two communications links: BLE (nrf52832 SoC) and UWB (DWM1000 transceiver). The nrf52832 SoC is built around an ARM Cortex-M4 architecture and provides 64kB of RAM and 512kB of ROM. The CPU can run at 64Mhz at maximum. The form-factor of the DWM1001-DEV development board

[1]https://www.iot-lab.info/docs/boards/overview/

[2]https://www.iot-lab.info/docs/getting-started/design

also includes an on-board debugger making it an ideal target for experimental deployment. Typically, the BLE interface can be used for commissioning the embedded application whilst the UWB interface is used for time-based ranging measurements and networking. Although the vendor provides factory firmware, the user can freely reprogram the module in order to fit his application's requirements. This feature, coupled with the use of the widely adopted nrf52 SoC, makes this module an ideal candidate for the experiments targeted by this testbed.

C. Open-Source Embedded Software Platform

To build the firmware embedded on the DWM-1001-DEV nodes, we have integrated open source building blocks combining:

- the real-time operating system RIOT [13];
- the uwb-core library[3] developed by one of the leading manufacturers of UWB indoor location products, which we have ported to RIOT;
- the NimBLE library [4], which provides Bluetooth Low-Energy network stack support, which as already supported in RIOT.

Note that on the hardware we provide on the platform, other open source building blocks may also be used to build firmwares supporting UWB [10] as well as BLE. Some take a `bare-metal` approach while others take advantage of the features offered by an RTOS (such as RIOT, MyNewt etc.) and are designed to be easily portable.

Depending on the application's requirements, the user can adopt different strategies for the firmware development. For instance, in order to develop precise Location-based services, one could use the PANS framework's APIs provided by the vendor which allows to extend the factory firmware with application-specific code, and thus taking benefit from existing services exposed by the framework such as: BLE commissioning from mobile phone, RTLS network (anchors, gateway, etc.).

Although this paradigm might be suitable for cases where the developer is a user of location services, it is inadequate for experimentation, research, etc.

IV. UWB and BLE Ranging Tutorial

Based on the setup described in Section III, we provide two reference applications relevant to contact-tracing: BLE RSSI scan and UWB two-way-ranging. For both applications, we label the nodes as follows:

- *initiator* node: a node performing proximity tracing with its neighbors,
- *neighbor* nodes: nodes broadcasting their presence and/or interacting with the initiator node on-demand.

All the experiment process (build, experiment management, data gathering and plot) is fully automated and can be easily reproduced or adapted for other purposes [14] thanks to the tools offered by FIT IoT LAB.

[3]https://github.com/Decawave/uwb-core

[4]https://github.com/apache/mynewt-nimble

A. BLE RSSI tracing

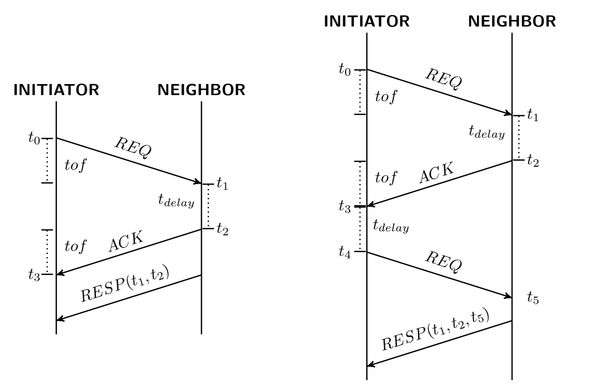
Based on the NimBLE stack, this application allows a BLE peripheral node to be configured, from serial command-line, in one of the following states:

- *Advertising* state: periodic transmit of a packet containing metadata such as node id and any application specific data[5].
- *Scanning* state: reception of advertisement packets with computation of the RSSI in *dBm*. Received packets and RSSI metrics are captured from the serial port and stored in *json* format.

Typically, a contact-tracing operation would submit such an experiment with one initiator node operating in *Scanning* state whilst its neighbor nodes are operating in *Advertising* state. The gathered data exposed by the testbed can then be analysed offline as discussed in Section V. The standard approach is to first fit a log-distance model to infer range between initiator and neighbor(s) from the captured RSSI of the link.

B. UWB ranging

The uwb-core library gives us access to different UWB services, including integrated support for SS-TWR, SDS-TWR, 2M-TWR, N-TWR. The library can also be extended with custom ranging protocols by implementing an internal low-level application interface.



(a) Single Sided TWR (b) Symmetric Double Sided TWR

Fig. 3. Subset of TWR protocol variants, 2M-TWR is similar to SS-TWR but ACK and RESP are combined into one message, N-TWR might be SDS or SS but against multiple neighbors. The neighbor response contains all required timestamps for the initiator to estimate the TOF.

By integrating uwb-core with RIOT we expose to the user a command-line application to perform range-requests following one of the above specified TWR protocols. The nodes can fulfill one of two roles:

- *tag*: initiates the TWR exchange by sending a range request.

[5]In BLE beaconing applications, the advertisement packet contains a reference power at a reference distance (eg. 1m) in order to compute the path-loss and thus infer the range to the sender based on predefined model.

- *anchor*: is constantly listening for range requests and sends a range request response.

Each device is subscribed to events marking the end of a two-way request. At that point, both devices know all the measured timestamps and from that can calculate the time-of-flight which can then be converted into range (distance) estimation. Other indicators such as RSSI, First Path Power Level (fppl), line-of-sight (los) likely-hood indicator [6] can also be recovered.

In a contact-tracing operation, a mobile device (initiator) would send range requests to its neighbors to keep track of those that have been in its near vicinity.

C. *Do ~~not~~ try this at home!*

We provide a step-by-step tutorial on how to perform UWB and BLE ranging on the testbed in our online guide [14].

Beyond our physical testbed, it is worth mentioning that the use of COTS hardware, open-source software and tools allows the users to run the same experiments in their own setups on larger scale or under different environmental settings : human presence, harsh environments, etc. In this sense, the experimental testbed serves not only for the solution development and evaluation, but also as as a benchmark for sharing and reproducing the experimental results and do not prevent to reproduce the embedded set up at home, quite the opposite. It is also clearly possible to use alternative firmware blocks such as MyNewt, FreeRTOS, Decawave's PANS or any framework [10] supporting the dwm1001 module.

V. Preliminary UWB & BLE Measurements

Based on the application described in Section IV, we can now collect experimental data in order to support the design of range-based applications such as contact-tracing. More precisely, we can plot the RSSI values of successive BLE scans and match them against the UWB time-based range estimates as well as ground-truth distances available from the testbed.

To this end, we reserve ten dwm1001 nodes that are located on the same plane and in geometric LOS conditions as depicted in Fig. 4. Using the testbed tools, we can remotely program the firmware, and via each node's serial interface, configure the nodes and capture the metrics of interest, namely: RSSI from BLE and time-based range estimates from UWB.

A. *Rssi-based range models*

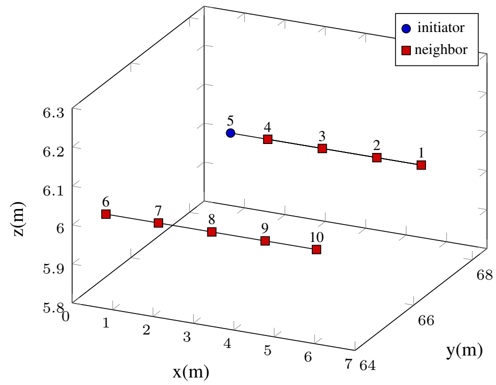
In order to perform range estimation form the RSSI, the standard approach is to identify the underlying log-distance path loss model [15], [16], [17]. Namely, the RSSI at a distance d is given by

$$\mathrm{RSSI}(d) = \mathrm{RSSI}(d_0) - 10 \times n \times \log_{10}\left(\frac{d}{d_0}\right), \qquad (1)$$

with the following model parameters:

- $\mathrm{RSSI}(d_0)$: the received power at a reference distance d_0, which is essentially hardware-dependent,

[6]DW1000 USER MANUAL, Section 4.7



(a) Position in 3D space

id	5	4	3	6	7	2	8	1	9	10
5	0.00	0.92	2.27	3.62	3.72	3.81	4.31	4.72	5.12	6.07
4	0.92	0.00	1.35	2.70	3.93	3.70	3.92	3.80	4.53	5.37
3	2.27	1.35	0.00	1.35	4.56	3.95	3.70	2.45	3.91	4.49
2	3.62	2.70	1.35	0.00	5.46	4.59	3.96	1.10	3.70	3.89
6	3.72	3.93	4.56	5.46	0.00	1.30	2.62	6.32	3.94	5.21
7	3.81	3.70	3.95	4.59	1.30	0.00	1.32	5.32	2.64	3.91
8	4.31	3.92	3.70	3.96	2.62	1.32	0.00	4.47	1.32	2.59
1	4.72	3.80	2.45	1.10	6.32	5.32	4.47	0.00	3.88	3.70
9	5.12	4.53	3.91	3.70	3.94	2.64	1.32	3.88	0.00	1.27
10	6.07	5.37	4.49	3.89	5.21	3.91	2.59	3.70	1.27	0.00

(b) Euclidean distances

Fig. 4. Selected nodes for the experiments. The nodes on the same line, that is the lines 1-5 and 6-10 are in good Line Of Sight (LOS) conditions, in opposition to the nodes on opposite rows since they are shaded by the wooden pillars 2.5x15 cm thick.

- n: the path-loss exponent, which is essentially environment-dependent.

For practical reasons, it is a common practice to set $d_0 = 1m$. Once the model parameters are identified, the range estimate is given by

$$\hat{d} = d_0 \times 10^{\dfrac{\mathrm{RSSI}(d_0) - \mathrm{RSSI}(d)}{10 \times n}}. \qquad (2)$$

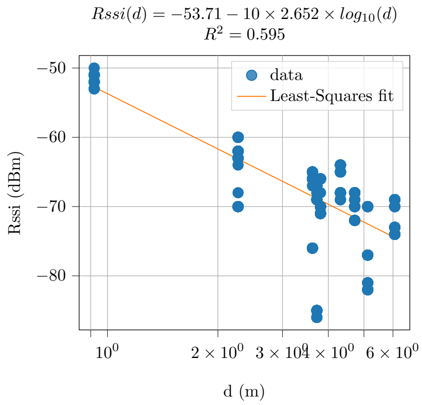
RSSI measurements between initiator node and its neighbors are available for both BLE and UWB channels. Based on such measurements, log-distance models can hence be identified to support range estimation. This is depicted by Fig. 5 and Fig. 6 along with corresponding modeling errors.

B. *Time-based* VS *Rssi-based range estimations*

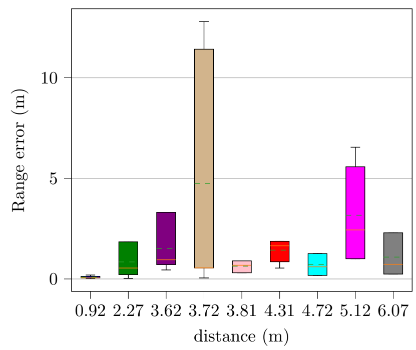
Based on the UWB ranging application, we can collect TWR range estimates and RSSI values between the initiator and its nine neighbors. From the log-distance model (Fig 6a), the RSSI values are then mapped to range estimates for comparison. On the other hand, based on the BLE RSSI tracing application, a similar comparison is conducted for BLE RSSI measurements thanks to the corresponding log-distance model (Fig 5a). The results are shown in Fig. 7. As expected, the UWB time-based ranging exhibits higher stability and precision characteristics for range estimates.

C. *Reproducing the results*

The results shown in this paper can be reproduced by following this guide [14]. Scripts are provided for automating



(a) Model curve fit

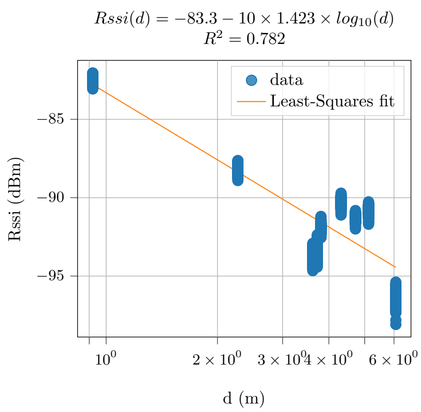


(b) Absolute modeling errors

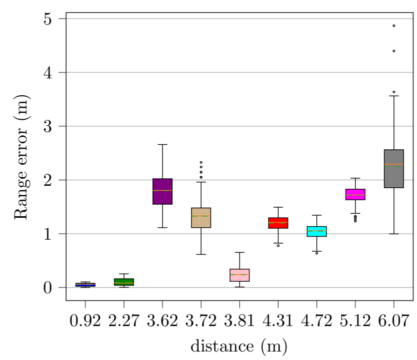
neighbors		range errors		
id	distance	mean	median	st. deviation
4	0.920	0.090	0.058	0.040
3	2.270	0.847	0.543	0.725
2	3.620	1.500	0.954	1.100
6	3.720	4.750	0.550	5.480
7	3.810	0.638	0.681	0.238
8	4.310	1.430	1.650	0.493
1	4.720	0.711	0.604	0.450
9	5.120	3.170	2.440	2.170
10	6.070	1.090	0.728	0.874

(c) Absolute modeling errors statistics

Fig. 5. BLE RSSI log-distance model fitting as seen from the initiator with 9 neighbors. The model was estimated from 100 RSSI measurements for each distance i.e a total of 900 samples.



(a) Model curve fit



(b) Absolute modeling errors

neighbors		range errors		
id	distance	mean	median	st. deviation
4	0.920	0.053	0.063	0.031
3	2.270	0.098	0.082	0.068
2	3.620	1.800	1.810	0.325
6	3.720	1.330	1.330	0.312
7	3.810	0.239	0.238	0.144
8	4.310	1.200	1.210	0.141
1	4.720	1.050	1.060	0.132
9	5.120	1.710	1.720	0.154
10	6.070	2.240	2.290	0.646

(c) Absolute modeling errors statistics

Fig. 6. UWB RSSI log-distance model fitting as seen from the initiator with 9 neighbors. The model was estimated from 100 RSSI measurements for each distance i.e a total of 900 samples.

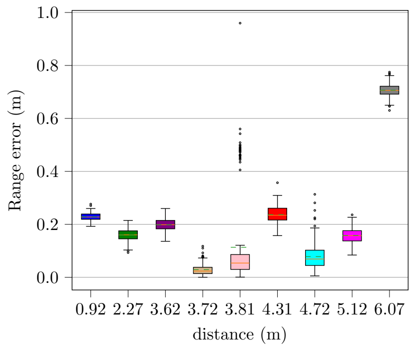
the experiment: build firmware, submit experiment, configure nodes, collect metrics. Based on the collected RSSI measurements, a specific script is provided to calibrate the RSSI models and save them for future use. Finally, other scripts are available for comparing range estimates against testbed's ground-truth distances, either from TWR or RSSI via the previously calibrated models.

VI. Next Steps & Future Work

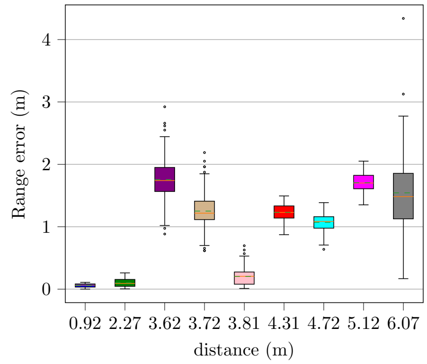
On one hand, we plan to leverage our platform to validate our research results on different novel proximity estimation techniques based on combining UWB and BLE. On the other hand, we plan on designing and comparatively evaluating additional on-board data preprocessing algorithms enabling anonymity and privacy-preserving contact event tracing (inspired by approaches such as DP3T, ROBERT, DESIRE etc. [18]). The evaluation will be performed in different environments and contexts such as featuring obstacles of different natures (walls, humans, etc) and with different patterns (fix, mobile, small, big, etc). Furthermore, our platform being purposely open-access, and our code being open source, we plan to foster use of this platform by the wider community of experimental researchers, and to collaboratively develop and maintain the implementations, upstream.

VII. Conclusion

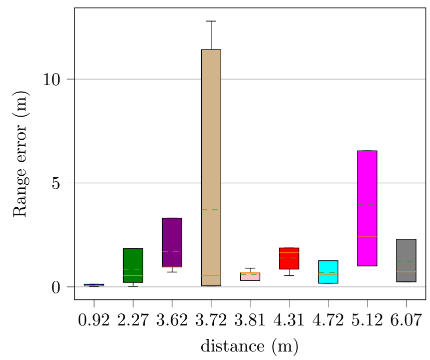
In this paper, we presented our current work on UWB and BLE based radiolocalization. We provide an open-access experimental platform, which we described in detail: set up, hardware, open-source software, experimental testbed, as well as a methodology and a step-by-step tutorial on how to use this platform. We then presented preliminary experimental results using this platform to produce UWB and BLE ranging measurements that are very encouraging and finally developed the different remaining steps and future work. We believe our contribution can serve as a solid basis for a large set of research on contact tracing and pave the way to new applications in this domain.



(a) UWB: TWR estimation errors



(b) UWB: RSSI path-loss model prediction errors



(c) BLE: RSSI path-loss model prediction errors

neighbors		range errors		
id	distance	mean	median	st. deviation
4	0.920	0.229	0.228	0.016
3	2.270	0.160	0.160	0.023
2	3.620	0.198	0.199	0.023
6	3.720	0.029	0.024	0.022
7	3.810	0.113	0.054	0.163
8	4.310	0.237	0.234	0.032
1	4.720	0.078	0.069	0.049
9	5.120	0.158	0.157	0.028
10	6.070	0.707	0.706	0.025

(d) UWB: TWR absolute estimation errors statistics

neighbors		range errors		
id	distance	mean	median	st. deviation
4	0.920	0.058	0.065	0.031
3	2.270	0.102	0.091	0.067
2	3.620	1.760	1.740	0.313
6	3.720	1.250	1.220	0.255
7	3.810	0.203	0.206	0.137
8	4.310	1.230	1.230	0.131
1	4.720	1.070	1.080	0.142
9	5.120	1.710	1.700	0.142
10	6.070	1.540	1.480	0.611

(e) UWB: RSSI path-loss model prediction errors statistics

neighbors		range errors		
id	distance	mean	median	st. deviation
4	0.920	0.087	0.058	0.037
3	2.270	0.839	0.543	0.692
2	3.620	1.700	0.954	1.160
6	3.720	3.710	0.550	5.180
7	3.810	0.603	0.681	0.233
8	4.310	1.390	1.650	0.482
1	4.720	0.697	0.604	0.446
9	5.120	3.950	2.440	2.420
10	6.070	1.230	0.728	0.898

(f) BLE: RSSI path-loss model prediction statistics

Fig. 7. Ranging performance for BLE and UWB links. For every initiator-neighbor link, 100 samples were collected for each metric (RSSI and TWR). Recalling that, as per Fig1, the neighbors 2-4 and 1 are in good LOS conditions with the initiator node (id=5) in opposition to the nodes 6-10.

ACKNOWLEDGMENT

The authors would like to thank the FIT IoT LAB project, and the RIOT community.

REFERENCES

[1] EIT News. (2020, May, 14) Anonymous covid-19 contact tracing using physical tokens. [Online]. Available: https://www.eitdigital.eu/newsroom/news/article/join-efforts-to-develop-covid-19-contact-tracing-using-physical-tokens/

[2] S. Alhadhrami *et al.*, "Ultra wideband positioning: An analytical study of emerging technologies," in *Proceedings of the Eighth International Conference on Sensor Technologies and Applications, SENSORCOMM*, 2014, pp. 1–9.

[3] D. Neirynck *et al.*, "An alternative double-sided two-way ranging method," in *2016 13th workshop on positioning, navigation and communications (WPNC)*. IEEE, 2016, pp. 1–4.

[4] Y. Jiang and V. C. Leung, "An asymmetric double sided two-way ranging for crystal offset," in *2007 International Symposium on Signals, Systems and Electronics*. IEEE, 2007, pp. 525–528.

[5] H. Kim, "Performance analysis of the sds-twr-ma algorithm," in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, 2009, pp. 399–403.

[6] Q. Dong and W. Dargie, "Evaluation of the reliability of rssi for indoor localization," in *2012 International Conference on Wireless Communications in Underground and Confined Areas*. IEEE, 2012, pp. 1–6.

[7] C. T. Nguyen *et al.*, "A comprehensive survey of enabling and emerging technologies for social distancing—part i: Fundamentals and enabling technologies," *IEEE Access*, vol. 8, pp. 153 479–153 507, 2020.

——, "A comprehensive survey of enabling and emerging technologies for social distancing—part ii: Emerging technologies and open issues," *IEEE Access*, vol. 8, pp. 154 209–154 236, 2020.

[8] N. Ahmed *et al.*, "A survey of covid-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134 577–134 601, 2020.

[9] O. Hahm *et al.*, "Operating Systems for Low-end Devices in the Internet of Things: a Survey," *IEEE Internet of Things Journal*, 2015.

[10] B. Van Herbruggen *et al.*, "Wi-pos: A low-cost, open source ultra-wideband (uwb) hardware platform with long range sub-ghz backbone," *Sensors*, vol. 19, no. 7, p. 1548, 2019.

[11] A. Van den Bossche *et al.*, "Locura: A new localisation and uwb-based ranging testbed for the internet of things," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2018, pp. 1–6.

[12] C. Adjih *et al.*, "FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 459–464.

[13] E. Baccelli *et al.*, "RIOT: An Open Source Operating System for Low-end Embedded Devices in the IoT," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4428–4440, 2018.

[14] R. Dagher *et al.* (2021, Jan.) Reproducing results of an open experimental platform for ranging, proximity & contact event tracking using ultra-wide-band & bluetooth low-energy. [Online]. Available: https://gitlab.inria.fr/pepper/cnert-2021

[15] H.-s. Cho *et al.*, "Accurate distance estimation between things: A self-correcting approach," *Open Journal of Internet Of Things (OJIOT)*, vol. 1, no. 2, pp. 19–27, 2015.

[16] A. Thaljaoui *et al.*, "Ble localization using rssi measurements and iringla," in *2015 IEEE international conference on industrial technology (ICIT)*. IEEE, 2015, pp. 2178–2183.

[17] K. E. Jeon *et al.*, "Ble beacons for internet of things applications: Survey, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 811–828, 2018.

[18] C. Castelluccia *et al.*, "Desire: A third way for a european exposure notification system leveraging the best of centralized and decentralized systems," Inria, https://hal.inria.fr/hal-02570382/document, Research Report 02570382, 2020.